UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/778,076	02/07/2001	Yutaka Haga	1046.1236/JDH	4573

21171 7590 07/12/2006

STAAS & HALSEY LLP
SUITE 700
1201 NEW YORK AVENUE, N.W.
WASHINGTON, DC 20005

EXAMINER	
YIGDALL, MICHAEL J	
ART UNIT	**PAPER NUMBER**
2192	

DATE MAILED: 07/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/778,076	HAGA, YUTAKA	
	Examiner	Art Unit	
	Michael J. Yigdall	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 May 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *8-17,19-28 and 30-42* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *8-17,19-28 and 30-42* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on May 9, 2006 has been entered. Claims 8-17, 19-28 and 30-42 are pending.

Response to Arguments

2. Applicant's arguments have been fully considered but they are not persuasive.

Applicant contends that the features of "identifying a type of said branch instruction by obtaining an instruction code from said branch source address and decoding said instruction code" and "when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated," as recited in claims 12, 23 and 34, are not discussed in the references (remarks, page 10). To support this argument, Applicant states that Smolders instead discusses that a "counter level tracing tool 31 saves the address of the beginning of the next basic block of code, which is the address where the interruption came from as shown in step 34" (column 4, lines 31-34), and not so as to provide a return address (remarks, page 10).

First, however, it is again noted that in Smolders, "the address of the beginning of the next basic block of code" is indeed a return address. Smolders expressly illustrates, "Next Block = Return Address" (step 34 in FIG. 3). The interrupt returns to "the address where the interruption came from." Second, and more importantly, it is not clear how this discussion in

Smolders supports Applicant's argument that the above features are not discussed in the references. These features were addressed with reference to Smolders and Yeh.

Applicant contends, "the Examiner has not supported his statement that Alexander's discussion of a periodic event, such as, for example, a timer interrupt is replaceable with a 'trace interrupt' as taught by Smolders" (remarks, page 11).

However, as Applicant acknowledges, Alexander expressly discloses that "although the depicted examples employ timer interrupts, other interrupts may be used to trigger the described sampling mechanism" (column 11, lines 23-25). One such interrupt is the trace interrupt of Smolders, which is generated after every branch instruction (see, for example, column 3, lines 58-61). Applicant apparently concludes that the trace interrupt of Smolders is not a "periodic event" in the context of Alexander. Again, however, the trace interrupt is generated after every branch instruction, which means it is a periodic event. Furthermore, as Applicant notes, Alexander discloses a "page fault interrupt" as another alternative to the timer interrupt (see, for example, column 8, lines 51-53). In other words, the timer interrupts are not the only "periodic events" that Alexander contemplates. If one may substitute the timer interrupts with page fault interrupts, as Alexander suggests, certainly one may also substitute the timer interrupts with other interrupts, as Alexander also suggests, such as with the trace interrupts of Smolders.

Applicant contends, presumably with reference to claims 40-42, that none of the references discuss "an executor managing table, a subroutine managing table, and a calling managing table" (remarks, page 12).

However, the Office action set forth a *prima facie* case that these tables would have been obvious to one of ordinary skill in the art. The rejection is not based solely on the knowledge of the examiner, but rather on Alexander's suggestion (column 6, lines 54-61):

> Those of ordinary skill [in the] art will appreciate that tree structure 500 may be implemented in a variety of ways and that many different types of statistics may be maintained at the nodes other than those in the depicted example. In addition, other pointers may be stored within the nodes to further aid subsequent analysis. Further, other structural elements, such as tables for properties of the routine, such as, for example, the name of the routine, also may be stored within a node.

Alexander illustrates tree structure 500 in FIG. 5. None of the tables recited in the claims are functionally distinct from Alexander's tree structure (see, for example, column 6, lines 37-53) and the other data structures in the reference (see, for example, FIG. 4A). In other words, the executor managing table, subroutine managing table and calling managing table are functionally the same as Alexander's data structures.

As indicated in the Office action, the references do not expressly disclose merely the recited names of the tables and the recited pointers to the tables. For example, as recited in the claims, the subroutine managing table stores a subroutine address, times of calling the subroutine, a cumulative execution time of the subroutine, and the last called time of the subroutine. Indeed, Alexander's data structures store these elements (see, for example, FIG. 5 and column 6, lines 37-53, and FIG. 4A). Alexander does not expressly name or label these elements the "subroutine managing table," and does not expressly disclose a "pointer to assign the subroutine managing table."

Nonetheless, pointers to Alexander's data structures are necessary to access the data structures. Alexander even suggests that "other pointers may be stored within the nodes to further aid subsequent analysis," as noted above. Furthermore, the names or labels given to the

tables, as recited in the claims, do not render them functionally distinct from Alexander's data structures. Therefore, absent any evidence of new or unexpected results, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include such tables in pointers in Alexander.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 8-17, 19-28 and 30-42 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,002,872 to Alexander, III et al. (art of record, "Alexander") in view of U.S. Patent No. 6,253,338 to Smolders (art of record, "Smolders") in view of U.S. Patent No. 6,427,206 to Yeh et al: (art of record, "Yeh").

With respect to claim 12 (currently amended), Alexander discloses an apparatus for collecting a profile of a subroutine included in a program (see, for example, the title and abstract), comprising:

(a) a storage unit storing a profile (see, for example, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, noting that each node is considered a storage unit);

(b) an analyzing section, when an interrupt is generated during execution of said program: obtaining a branch source address and a branch destination address from a source of said interrupt (see, for example, column 5, lines 20-32, which shows analyzing the stack frames in response to an interrupt to identify subroutines, and column 5, lines 41-62, which shows obtaining a call or branch source address and a return or branch destination address).

Although Alexander discloses generating timer interrupts (see, for example, column 4, lines 20-23), and suggests that other interrupts may be generated instead (see, for example, column 11, lines 22-25), Alexander does not expressly disclose the limitation wherein the interrupt is generated by execution of a branch instruction, and Alexander does not expressly disclose identifying a type of said branch instruction by obtaining an instruction code from said branch source address and decoding said instruction code.

However, Smolders discloses generating an interrupt by execution of a branch instruction (see, for example, column 3, lines 58-61). Smolders further discloses an instruction flow unit that dispatches instructions to selected execution units for execution (see, for example, column 3, lines 10-13). The execution units include fixed-point execution units, load/store execution units, and floating-point execution units (see, for example, column 3, lines 13-15). The instruction flow unit cannot dispatch instructions in this manner without first interpreting or "decoding" the instruction codes, so as to determine the appropriate execution unit. Furthermore, it is understood in the art that such instruction decoding is an integral part of the instruction cycle.

Thus, Smolders discloses a system for collecting a trace of a program (see, for example, the title and abstract), wherein an interrupt is generated by execution of a branch instruction, and wherein the type of the branch instruction is inherently identified by obtaining an instruction

code and decoding the instruction code. The system enables tracing without introducing any overhead and without modifying the code (see, for example, column 1, lines 64-67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the profiling system of Alexander with the features taught by Smolders and to substitute the timer interrupt of Alexander with the branch interrupt of Smolders, as suggested by Alexander, so as to obviate any overhead and modifications to the code.

Moreover, Yeh expressly discloses obtaining an instruction code and decoding the instruction code (see, for example, column 3, lines 45-49), and identifying the type branch, such as whether the branch is a calling instruction or a return instruction (see, for example, column 3, lines 63-65), so as to collect branch predictions and provide prediction hints (see, for example, column 3, lines 52-60). The branch prediction system of Yeh enhances profiling (see, for example, column 2, lines 1-13) to enable speculative execution of instructions (see, for example, column 1, lines 29-40).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the profiling system of Alexander with the features taught by Yeh and to identify the type of branch, as taught by Yeh, so as to collect branch predictions and enhance the collection of profiles for purposes of speculative execution.

Alexander in view of Smolders in view of Yeh further discloses:

(c) a collecting section: obtaining said branch source address, said branch destination address, and an identified result from said analyzing section when the identified instruction is a calling instruction or a return instruction of said subroutine (see, for example, Alexander, column 5, lines 41-62, which shows obtaining a call or branch source address and a return or branch

destination address for a subroutine); and when said identified result is said calling instruction, storing said branch destination address as a subroutine address corresponding to said calling instruction and a calling time of said subroutine corresponding to said calling instruction in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address and a base time), when said identified result is said return instruction, obtaining a return time of said subroutine corresponding to said return instruction, calculating a execution time of said subroutine based on said obtained return time and said calling time, and storing a cumulative value of said execution time as said profile in correspondence with said branch destination address in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including an execution time and a cumulative time, inherently calculated based on the calling time and the return time, and see, for example, FIG. 4A, which shows timestamps for entering and returning from subroutines), and when the identified branch instruction is neither a calling instruction nor a return instruction, said interrupt is terminated (see, for example, Yeh, step 310 in FIG. 3 and column 8, lines 42-48, which shows not collecting history information when the branch is not a calling instruction or a return instruction, and see, for example, Smolders, steps 46 and 64 in FIG. 3 and column 5, lines 43-52, which shows terminating the interrupt when trace information is not to be collected).

With respect to claim 8 (previously presented), the rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein a plurality of storage units respectively corresponding to a plurality of executors of said subroutine

are prepared (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a plurality of nodes or storage units); and

said collecting section specifies said executor of said subroutine and stores said profile of said subroutine corresponding to said specified executor in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles; note the parent node pointers, which specify the executors of subroutines).

With respect to claim 9 (original), the rejection of claim 8 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section individually stores profiles of a plurality of subroutines corresponding to a specified executor in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, noting that a path in the tree represents the subroutines executed by a specific executor).

With respect to claim 10 (original), the rejection of claim 9 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section individually stores a first profile of a subroutine called by a main routine, and a second profile of the subroutine called by another subroutine, in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, and FIG. 10 and column 8, lines 24-40, which shows individual profiles of subroutines organized based on the calling routine, and see, for example, FIG. 8, which shows a subroutine Y called by both the main routine and by a second routine X).

With respect to claim 11 (original), the rejection of claim 10 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section stores said second profile and calling relationship information relating to said second profile, said calling relationship information indicating a relationship between said other subroutine and said called subroutine, in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including calling relationship information).

With respect to claim 13 (previously presented), the rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section stores times of calling of said subroutine corresponding to said branch destination address as said profile in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address and a base time, and see, for example, FIG. 4A, which shows timestamps for entering, i.e. times of calling, and returning from subroutines).

With respect to claim 14 (original), the rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section obtains an overhead of said subroutine as said profile and stores said overhead in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a time consumed by a thread executing a subroutine, which is considered a measure of overhead).

With respect to claim 15 (previously presented), the rejection of claim 13 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section, when said identified result is said calling instruction, stores an identifier of an executor of said subroutine corresponding to said calling instruction and said branch destination address in said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address, and note the parent node pointers, which identify the executors of subroutines).

With respect to claim 16 (previously presented), the rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein said collecting section, when said identified result is said calling instruction and said branch source address and said branch destination address are addresses of said subroutines, stores said branch source address and branch destination address as calling relationship information indicating a callings source subroutine and a calling destination subroutine is said storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the data structure used to store the profiles, including a subroutine address and calling relationship information), and stores at least one of the cumulative execution time and the times of calling in said calling destination subroutine in the call source subroutine, as said profile corresponding to said calling relationship information, in said storage unit (see, for example, Alexander, column 6, lines 37-53, which shows that the profile comprises both a base time and a cumulative execution time).

With respect to claim 17 (previously presented), the rejection of claim 12 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses a setting section setting an

execution environment of a source of said interrupt so as to generate said interrupt when said branch instruction is executed during the execution of said program (see, for example, Smolders, column 3, lines 58-61, which shows setting the execution environment to generate a trace interrupt after every branch instruction).

With respect to claim 23 (currently amended), the computer readable medium recited in the claim is analogous to the apparatus recited in claim 12 (see the rejection of claim 12 above, and see, for example, Alexander, column 11, lines 6-17, which shows a computer readable medium).

With respect to claims 19-22 and 24-28 (previously presented), the limitations recited in the claims are analogous to those of claims 8-11 and 13-17, respectively (see the rejection of claims 8-11 and 13-17 above).

With respect to claim 34 (currently amended), the method recited in the claim is analogous to the apparatus recited in claim 12 (see the rejection of claim 12 above).

With respect to claim 30-33 and 35-39 (previously presented), the limitations recited in the claims are analogous to those of claims 8-11 and 13-17, respectively (see the rejection of claims 8-11 and 13-17 above).

With respect to claim 40 (currently amended), the rejection of claim 13 is incorporated, and Alexander in view of Smolders in view of Yeh further discloses the limitation wherein the collecting section generates a control table corresponding to each executor of the subroutine on the storage unit (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows the

data structure used to store the profiles; note that a path in the tree represents the subroutines executed by a specific executor, and see, for example, FIG. 9 and column 8, lines 24-33, which shows the data structure in the form of a table),

wherein the control table includes an executor managing table, a subroutine managing table, and a calling managing table (see, for example, Alexander, column 6, lines 54-61, which shows that the data structure may include other pointers and tables to aid in subsequent analysis).

wherein the executor managing table stores an identifier of the executor and a pointer to assign the subroutine managing table (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows parent node pointers that identify the executors of subroutines),

wherein the subroutine managing table is generated for every subroutine executed by the executor, the subroutine managing table storing a subroutine address, times of calling of the subroutine, a cumulative execution time of the subroutine, the last called time of the subroutine, and a pointer to assign the calling managing table (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows a subroutine address, a base time and a cumulative time, and see, for example, FIG. 4A, which shows timestamps for entering and returning from subroutines), and

wherein the calling managing table is generated for every subroutine called by the subroutine, the calling managing table storing a branch source address as a calling subroutine address, a branch destination address as a called subroutine address, times of calling of the called subroutine, a cumulative execution time of the called subroutine, the last called time of the called subroutine, and a pointer to specify the subroutine managing table managing the calling subroutine (see, for example, Alexander, FIG. 5 and column 6, lines 37-53, which shows a

subroutine address, a base time and a cumulative time, and FIG. 4A, which shows timestamps for entering and returning from subroutines, and see, for example, column 5, lines 41-62, which shows a call or branch source address and a return or branch destination address for a subroutine).

Although Alexander in view of Smolders in view of Yeh does not expressly disclose the recited table names and pointers, Alexander in view of Smolders in view of Yeh discloses that the data structure may include such pointers and tables to aid in subsequent analysis (see, for example, Alexander, column 6, lines 54-61). Alexander in view of Smolders in view of Yeh further discloses the recited information stored in the data structure, as presented above.

Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to supplement the data structure of Alexander in view of Smolders in view of Yeh with other pointers and tables, as suggested by Alexander, including an executor managing table, a subroutine managing table, and a calling managing table, each with corresponding pointers, for the purpose of facilitating subsequent analysis of the profiles.

With respect to claims 41 and 42 (currently amended), the limitations recited in the claims are analogous to those of claim 40 (see the rejection of claim 40 above).

Conclusion

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (571) 272-3707. The examiner can normally be reached on Monday through Friday from 7:30am to 4:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MY

Michael J. Yigdall
Examiner
Art Unit 2192

mjy

TUAN DAM
SUPERVISORY PATENT EXAMINER